A Laparoscopic Liver Navigation Pipeline with Minimal Setup Requirements

Reuben Docea[a,†], Micha Pfeiffer[a,†], Jan Müller[b], Katja Krug[b], Matthias Hardner[b], Paul Riedel[b], Martin Menzel[a], Fiona R. Kolbinger[a,c,d], Laura Frohneberg[c], Jürgen Weitz[c,d,e] and Stefanie Speidel[a,d,e]

[†]Equal contribution
[a]National Center for Tumor Diseases (NCT), Dresden, Germany
[b]Technische Universität Dresden, Germany
[c]Faculty of Medicine and University Hospital Carl Gustav Carus, Technische Universität Dresden, Germany
[d]Else Kröner Fresenius Center for Digital Health Dresden, Technische Universität Dresden, Germany
[e]Centre for Tactile Internet with Human-in-the-Loop, Technische Universität Dresden, Germany
E-mail: (reuben.docea, micha.pfeiffer)@nct-dresden.de, (jan.mueller, katja.krug, matthias.hardner, paul.riedel)@tu-dresden.de

***Abstract*—In the case of liver tumor resections, Minimally Invasive Surgery (MIS) poses several benefits over open surgery. However, MIS makes navigating the surgical scene considerably more challenging, which hampers the realization of its full potential. Many Image Guidance Navigation Systems (IGS) have been proposed to overcome these challenges. The majority of these depend on optical tracking systems, whose additional setup overhead is a barrier to clinical translation. In this paper we put forward an IGS prototype which eliminates the need for optical tracking, and additionally incorporates a user-oriented camera calibration method which is more reliable and faster than typical checkerboard methods. We lastly make publicly available the core system modules for 3D reconstruction and rigid registration.**

***Index Terms*—Augmented reality, Computer assisted navigation, Computer assisted surgery, Image guided surgery, Laparoscopic liver resection, Machine vision, Robotic liver surgery, Robot Operating System**

I. Introduction

In recent years, liver tumor resections have moved increasingly in the direction of being performed through small incisions with long narrow instruments. This is an instance of Minimally Invasive Surgery (MIS), known as laparoscopic liver resection (LLR). The shift is owed to the benefits MIS poses over open surgery, such as decreased morbidity and improved cost-effectiveness [1]. Despite promising clear advantages, a number of factors make LLR more difficult to perform by complicating the navigation of the surgical scene, consequently hindering the realization of its full potential. Among these are the loss of the ability to palpate the liver for tumors, a requirement for skilled hand-eye coordination, and the lack of depth perception [2]. To mitigate these issues and better facilitate LLR, a number of Image Guidance Navigation Systems (IGS) have been developed. These aim to assist the surgeon by enabling the intraoperative visualization of relevant anatomical structures such as vasculature and tumors. IGS approaches may be categorized by the input data they are based on: video, ultrasound, Computed Tomography (CT), and Magnetic Resonance Imaging [2].

In this work we present a *video-based* IGS that delivers improved usability from the surgeon's perspective and is collaboratively built using the Robot Operating System (ROS) framework. Of the V-IGS systems put forward in recent work, many rely on external optical tracking to obtain the pose of the laparoscope camera [3]–[5]. This often involves the attachment of trackable markers to the laparoscope and carrying out a hand-eye calibration to obtain the spatial relationship between the marker and the laparoscope cameras. In tandem with this, additional optically tracked devices are often used in the process of registration, either to manually intraoperatively register the preoperative CT liver model in the first instance [6], or to collect intraoperative liver surface points with which to then register the preoperative model [7]. The additional setup time and equipment which these demand is a barrier to use in the operating room.

Our pipeline, whose workflow is detailed below, overcomes the requirement for an optical tracking system by using simultaneous localization and mapping (SLAM) to track the pose of the laparoscope camera, together with a fully software-based registration method. An additional advantage of our approach is that a flexible endoscope may be used, allowing the method to transfer to other scenarios. A second stand-out aspect of our pipeline is the incorporation of a camera calibration method that is faster and more reliable than typical methods, which features a 3D calibration field and an application that provides user-friendly guidance. The innovations within the IGS we put forward bring us closer the goal of truly translating such a system to the operation room. Alongside this publication, we make available two of the core modules of the system: *Point Cloud Fusion* and *Semi-Automatic Registration* (see footnote of first page).

The authors gratefully acknowledge funding for this research by the State of Saxony via Sächsische Aufbaubank (SAB) in the scope of the ARAILIS project (100400076). This measure is co-financed with tax funds on the basis of the budget passed by the Saxon state parliament.

Project page: https://gitlab.com/arailis-public/laparoscopic-liver-navigation

Workflow Description

Prior to the operation, a CT scan of the liver is obtained and segmented to identify structures of interest such as liver, vasculature, and tumors. These data are used to create a preoperative 3D model. Due to the instability of laparoscope camera parameters [8], the first step in the operating room before surgery is an accuracy check with the calibration module. If the system error is above a threshold of acceptability, the laparoscope camera must be re-calibrated.

At this point, the surgeon may build the 3D reconstruction of the liver by surveying the liver surface with the laparoscope. Progressively, the intraoperative surface of the liver is reconstructed. Once this process is complete, the semi-automatic registration tool can be used to identify common areas on the preoperative and intraoperative models. Using these regions, the registration method will produce a rough alignment of the two models, which may then be refined by the non-rigid registration module.

Throughout, the process is conducted and visualized with the assistance of the available User Interface (UI) control and visualization tools.

II. System Design

In this section, we provide detail regarding the modules in the Pipeline outlined in Fig. 1.

A. Camera Calibration

A necessary prerequisite to using a camera system for 3D measurements is calibration, where the mathematical parameters describing the camera are determined prior to the operation. Laparoscope design does not prioritize stability, which affects how quickly accuracy declines over time and how often calibration must be repeated.

Our choice of method considers the medical environment, where calibration must be performed by medical personnel in challenging conditions and with limited time. The selected method is therefore one that is simple to use, fast, and requires no in-depth understanding of camera calibration. This is achieved by real-time detection of a 3D calibration field and feedback to the user on the optimal capture of necessary images. The information is presented as an overlay of the laparoscope image, illustrated in Fig. 2. A background process continuously checks if the requirements are met for the image to be further used in the accuracy check or calibration. A more detailed description of the method can be found in [8].

Within 10 to 20 seconds the accuracy check determines if the current camera parameters are still valid or need to be updated. This check requires images of the calibration field to be captured at a predefined distance. Using the known coordinates of the calibration field, an accuracy value is calculated. By comparison to a fixed threshold, a recommendation is displayed to the user if re-calibration is necessary.

For the calibration procedure, images are captured from multiple previously defined camera positions to which the user is guided one by one. With these, the calibration is calculated in the background. Using the new parameters, the result of a new accuracy check is compared to the pre-calibration accuracy. If the accuracy is acceptable and has improved, a recommendation to save the parameters is given.

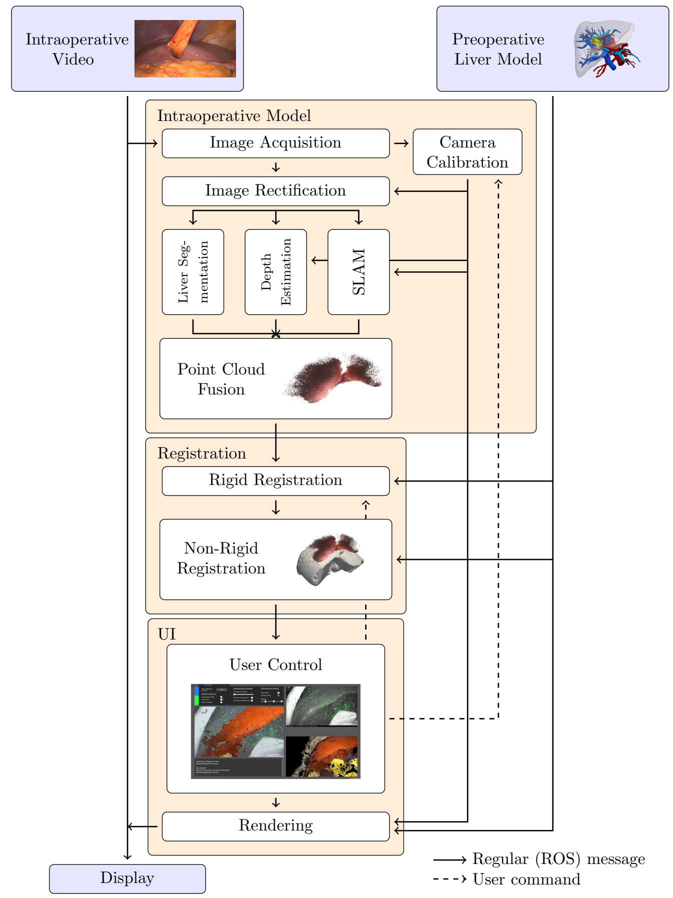

Fig. 1. Pipeline overview. The preoperative liver model is fused into the intraoperative video stream. For this, a 3D point cloud is reconstructed from the intraoperative video stream (intraoperative model). Then, the preoperative model is aligned with this intraoperative model and rendered from the current camera perspective before being overlayed with the video stream.

B. SLAM

In order to obtain poses for the camera, our system employs the popular SLAM module ORB-SLAM2 [9], [10]. This is a sparse, keyframe-based SLAM system, whose name derives from the ORB features [11] it uses. Bundle adjustment is used to compute the camera trajectory, and the incorporated Bag-of-Words method DBoW2 [12] enables re-localization when tracking is lost. Running on four threads, it achieves stable and accurate tracking on rectified stereo image pairs from the laparoscope.

C. Disparity and Depth Estimation

For the 3D reconstruction of the liver, a dense depth map of the scene is required. Depth information is acquired by stereo disparity estimation on rectified image pairs. We employ the Hierarchical Stereo Matching (HSM) network [13], a neural network that searches for stereo correspondences in a coarse-to-fine hierarchy. This method delivers high-quality *dense*

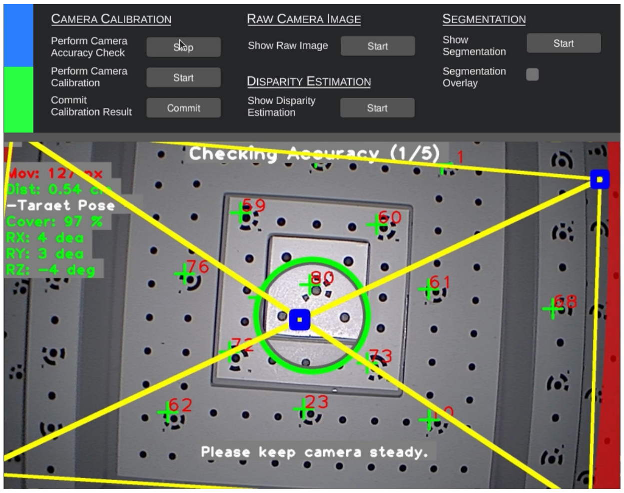

Fig. 2. Screenshot of calibration module within Unity UI, capturing images of the 3D calibration field for accuracy checking.

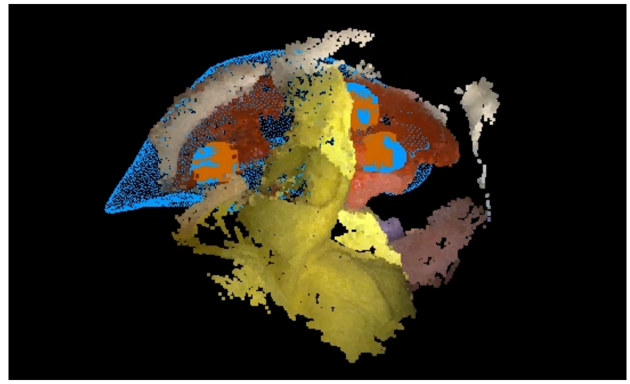

Fig. 3. Rendering of rigidly registered preoperative (blue) and intraoperative models.

disparity, similar to state-of-the-art methods, while being faster than comparable approaches.

For acceptable throughput, initial versions of our system processed images downscaled to 1/16 of the laparoscope's native FullHD resolution. To benefit from the full resolution of images, and hence of disparity and depth results, we put effort into accelerating the HSM algorithm. We applied the network inference acceleration technique TensorRT [14], and a pipelining of the pre-processing step using the DALI library [15], both utilizing the CUDA parallel programming model for GPUs. In addition, we designed a multithreaded processing scheme that allows parallel execution of several CPU threads and the usage of multiple GPUs. As a result, we are now able to produce FullHD high-quality disparity maps at the laparoscope's full rate of 30 frames/s.

D. Segmentation

To separate depth and texture information of the liver from other abdominal organs and surgical instruments, an automatic image segmentation is applied to the rectified images. Initially using a UNet/Ternausnet binary segmentation network [16], we added dropout [17] and a min-max normalization of all color channels [18]. To better suit the system requirements, we newly trained the network with intraoperative images.

The original inference implementation using PyTorch runs at almost full framerate, but exhibits very high CPU load. To achieve maximum speed at relatively low CPU load, we restructured the module, relocating the pre-processing to the GPU, running inference directly on the results of this step, and afterwards keeping the data on the GPU for post-processing.

E. Point Cloud Fusion

Natively, ORB-SLAM2 produces only a sparse map of the surveyed environment based on detected salient points. In our pipeline we implemented an extension for dense output as recommended by the ORB-SLAM2 authors [9], [10]. For each coincident set of camera pose, estimated depth and segmentation, a point cloud consisting of points belonging to the liver is created by projecting the left camera image into object space. With respect to the hue and position of each point, a counter is updated to record the repeated observance of points. Once the count for a point exceeds a threshold, it is included in the dense output, finally producing a dense map. Previously recorded points, which are no longer visible in the view frustum, are removed, as they are likely the result of noise or deformation.

F. Registration

The core of a surgical navigation system is the fusion of the preoperative data into the intraoperative scene. In our case, the goal is to align the 3D liver model extracted from the CT scan with the dense reconstruction of the current scene, as depicted in Fig. 3. Since there is no fully automatic, real-time non-rigid liver registration method to date, our registration system is comprised of a semi-automatic rigid alignment, followed by a fast non-rigid alignment correction.

The initial rigid alignment is required to bring the preoperative and intraoperative models into rough alignment. We use a region-based registration method which attempts to find a rigid transform that aligns user-defined matching regions on the pre- and intraoperative liver surfaces [19]. To select these regions, we implement a user-friendly brush-select tool. The algorithm then uses a weighted iterative closest point method to calculate the required transform. The alignment procedure takes a trained user only a short time and usually has to be performed only once.

This is followed by a non-rigid alignment method which deforms the preoperative liver mesh in such a way that it aligns well with the intraoperatively recorded dense point cloud. We use the V2S-Net [20] for this step, a deep neural network which was trained on synthetic deformation data to compute a displacement field between the preoperative organ model and the partial, noisy intraoperative point cloud. This network is triggered every time the dense reconstruction is updated, usually multiple times a second. The geometry of internal vessels and tumors can then be deformed according to the estimated displacement field.

G. *User Interface*

The User Interface of our pipeline serves as a central application for gaining insight into the most relevant processes, such as the calibration, depth estimation, segmentation and the aligned preoperative model. This facilitates easily accessible control of visualization parameters across modules and enables real-time rendering. The underlying processes and transmitted messages can be visualized via ROS standard tools.

The UI is implemented in C# via the cross-platform runtime environment *Unity*. Two main modes, preparation and alignment, form the UI. In *preparation* mode, the UI visualizes the camera calibration, image segmentation and depth estimation. The segmented image, for example, can be overlayed onto the raw camera image to evaluate the segmentation results. In *alignment* mode, the largest image pane shows a 3D scene which combines the dense map and the correctly aligned preoperative model. Furthermore, an additional overlay of the camera image with the aforementioned scene objects can be shown. What is visualized in this main scene is highly customizable in terms of transparency and model slicing, which empowers the surgical team with a wealth of possibilities to secure a strong scene understanding.

III. Implementation

A. *ROS Integration*

The presented system is implemented as a modular architecture based on the communication framework ROS [21]. Every part shown in Fig. 1 represents a module or node in the ROS framework. The communication between the modules is established with ROS via *subscriber/publisher* (sub/pub) transfers, comparable to the UDP protocol, and *service* calls as with the TCP protocol. The sub/pub communication is used to transmit all data, such as camera images. We utilize the service methods of the ROS framework to establish control interactions between modules. For example, to control the calibration process through the UI, the user may start and stop the calibration or change the mode of operation.

As the user interface is implemented with the development platform Unity, a special ROS interface to Unity is required. To this end, the Robotics Hub [22] establishes the communication to the ROS-based modules using a websocket.

B. *Hardware Environment*

Implementation and test of our system are carried out in an Experimental Operation Room environment using an Aesculap EinsteinVision 3.0 FullHD stereo laparoscope [23] on a human body phantom. We run the prototype using two computers: a *back-end* workstation with two Intel Xeon 4216 CPUs and four NVIDIA A5000 GPUs, and a *front-end* workstation with an Intel i9-10900X CPU and an NVIDIA RTX2080 GPU.

For the implementation of the complete processing chain we developed a distributed-computing model that runs front-end tasks (image recording, calibration, user interface, etc.) on the front-end computer, while computationally demanding tasks are placed on the back-end workstation. With ROS handling the networking protocols, interfaces were developed for fast (10 GbE) and ”normal” network connections (1 GbE, requiring compression/decompression of the transported data).

IV. Discussion & Conclusion

Many existing IGS are burdened by the additional setup time and hardware of an optical tracking system. Despite eliminating optical tracking, the presented pipeline reliably achieves good registrations (see Fig. 3). We find this to be the case for both phantom data and, retrospectively, for data from human surgeries. For the semi-automatic registration process, 2-3 minutes are required. Subsequent non-rigid updates then occur automatically, multiple times a second, without additional user input. Furthermore, the calibration can be performed in under 2 minutes, and is more accessible to non-expert users thanks to the implemented guidance method. Our system advances on previous approaches, principally by doing away with external tracking. Regarding time constraints, our system better meets clinical requirements by shortening the setup process. With these improvements, our approach simplifies the setup, and additionally enables the use of flexible endoscopes. Ultimately, these aspects significantly lower existing barriers to clinical translation. However, an assessment of the final registration accuracy is desirable. While difficult to obtain *in-vivo* due to the procedure's invasive nature, it is feasible using a Human phantom model and a CT scanner. This aspect is to be considered in future work.

Additionally, there are various ways in which the pipeline could be built upon. Aside from increasing the accuracy of all system components, one area for improvement is the SLAM system. While effective for this application in relative terms, it fails to track the camera's position when the surgeon moves the laparoscope very quickly, and consequently stops building the map. Other SLAM systems aim to deal with this issue, such as ORB-SLAM3 [24]. Yet, in in-lab assessments ORB-SLAM3 did not achieve a framerate comparable to ORB-SLAM2.

ROS serves well as a platform for collaborative development of the pipeline, by simplifying the integration of separately developed modules and thus allowing each partner to focus on their contribution. However, there remain speed limitations such that when transmitting point cloud data using the framework, we cannot always benefit from the full disparity rate. Overcoming this issue would enable the 3D reconstruction to operate even more smoothly.

While many improvements remain to be made before an IGS can comfortably be used throughout surgery, the system put forward here represents a large step in this direction.

Acknowledgment

We would like to express our gratitude to the medical personnel from the Department for Visceral, Thoracic and Vascular Surgery of University Hospital Carl Gustav Carus, Dresden, whose provision of data and feedback were essential throughout the development of the pipeline.

References

[1] A. A. Fretland, D. Aghayan, and B. Edwin, "Long-term survival after laparoscopic versus open resection for colorectal liver metastases." *Journal of Clinical Oncology*, vol. 37, no. 18_suppl, pp. LBA3516–LBA3516, 2019. [Online]. Available: https://doi.org/10.1200/JCO.2019.37.18_suppl.LBA3516

[2] C. Schneider, M. Allam, D. Stoyanov, D. Hawkes, K. Gurusamy, and B. Davidson, "Performance of image guided navigation in laparoscopic liver surgery - a systematic review," *Surgical Oncology*, vol. 38, p. 101637, 2021.

[3] J. S. Heiselman, L. W. Clements, J. A. Collins, J. A. Weis, A. L. Simpson, S. K. G. M.D., T. P. K. M.D., W. R. J. M.D., and M. I. Miga, "Characterization and correction of intraoperative soft tissue deformation in image-guided laparoscopic liver surgery," *Journal of Medical Imaging*, vol. 5, no. 2, pp. 1 – 12, 2017. [Online]. Available: https://doi.org/10.1117/1.JMI.5.2.021203

[4] G. A. Prevost, B. Eigl, I. Paolucci, T. Rudolph, M. Peterhans, S. Weber, G. Beldi, D. Candinas, and A. Lachenmayer, "Efficiency, accuracy and clinical applicability of a new image-guided surgery system in 3D laparoscopic liver surgery," *Journal of Gastrointestinal Surgery*, vol. 24, no. 10, pp. 2251–2258, 2020.

[5] C. Schneider, S. Thompson, J. Totz, Y. Song, M. Allam, M. Sodergren, A. Desjardins, D. Barratt, S. Ourselin, K. Gurusamy *et al.*, "Comparison of manual and semi-automatic registration in augmented reality image-guided liver surgery: a clinical feasibility study," *Surgical Endoscopy*, vol. 34, no. 10, pp. 4702–4711, 2020.

[6] S. Thompson, J. Totz, Y. Song, S. Johnsen, D. Stoyanov, S. Ourselin, K. Gurusamy, C. Schneider, B. Davidson, D. Hawkes, and M. J. Clarkson, "Accuracy validation of an image guided laparoscopy system for liver resection," in *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, R. J. W. III and Z. R. Yaniv, Eds., vol. 9415, International Society for Optics and Photonics. SPIE, 2015, pp. 52 – 63. [Online]. Available: https://doi.org/10.1117/12.2080974

[7] T. P. Kingham, S. Jayaraman, L. W. Clements, M. A. Scherer, J. D. Stefansic, and W. R. Jarnagin, "Evolution of image-guided liver surgery: transition from open to laparoscopic procedures," *Journal of Gastrointestinal Surgery*, vol. 17, no. 7, pp. 1274–1282, 2013.

[8] M. Hardner, R. Docea, and D. Schneider, "Guided calibration of medical stereo endoscopes," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2022, pp. 679–686, 2022. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2022/679/2022/

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[12] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[13] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical deep stereo matching on high-resolution images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5510–5519.

[14] NVIDIA, "NVIDIA TensorRT documentation," https://docs.nvidia.com/deeplearning/tensorrt/, online, accessed 2022-06-01.

[15] ——, "NVIDIA data loading library (DALI)," https://developer.nvidia.com/dali/, online, accessed 2022-06-01.

[16] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 encoder pre-trained on ImageNet for image segmentation," 2018. [Online]. Available: https://arxiv.org/abs/1801.05746

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] Baeldung, "Normalizing inputs for an artificial neural network," https://www.baeldung.com/cs/normalizing-inputs-artificial-neural-network, October 2020, online, accessed 2022-06-01.

[19] L. W. Clements, W. C. Chapman, B. M. Dawant, R. L. Galloway Jr., and M. I. Miga, "Robust surface registration using salient anatomical features for image-guided liver surgery: Algorithm and validation," *Medical Physics*, vol. 35, no. 6Part1, pp. 2528–2540, 2008. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2911920

[20] M. Pfeiffer, C. Riediger, S. Leger, J.-P. Kühn, D. Seppelt, R.-T. Hoffmann, J. Weitz, and S. Speidel, "Non-rigid volume to surface registration using a data-driven biomechanical model," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds. Cham: Springer International Publishing, 2020, pp. 724–734.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, January 2009.

[22] *Unity Technologies*, "Unity robotics hub," https://github.com/Unity-Technologies/Unity-Robotics-Hub, online, accessed 2022-06-01.

[23] B. Braun SE, "EinsteinVision 3D camera system," https://www.bbraun.co.uk/en/products/b50/einstein-vision.html, online, accessed 2022-06-01.

[24] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.